SECTION I—CLAIMS

Amendment to the Claims:

This listing of the claims will replace all prior versions and listings of claims in the application. Claim 35 is amended herein. No claims are canceled. No new claims are added. Claims 1-46 remain pending in the application.

Listing of Claims:

1. (Previously presented) A method for processing a group of instructions, the method comprising:

selecting an instruction to schedule, the instruction associated with an operand, wherein the operand comprises a live range representing the operational duration of the operand relative to the group of instructions being processed;

attempting to allocate a preserved register to the operand for its live range when its live range spans a function call;

attempting to allocate a scratch register to the operand for its live range when its live range does not span the function call; and

scheduling the instruction when the allocation of the preserved register or the scratch register is successful.

2. (Previously presented) The method of claim 1, further comprising:

terminating the attempt to allocate the preserved register or the scratch register to the operand for its live range when the operand's use by the selected instruction falls at the end of the live range; and marking the preserved register or the scratch register allocated to the operand as available for reallocation.

3. (Previously presented) The method of claim 1, wherein attempting to allocate the preserved register comprises:

allocating the preserved register to the operand for its live range when the preserved register is in a list of previously used preserved registers marked as available for allocation.

4. (Previously presented) The method of claim 3, further comprising:

attempting to find a second preserved register in the list of previously used preserved registers and currently allocated to a tentative live range of a second operand; and

reallocating the second preserved register to the operand for its live range and reserving a different preserved register to the second operand for its tentative live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range is successful.

5. (Previously presented) The method of claim 4, further comprising

adding a new preserved register to the list of previously used preserved registers; and

allocating the new preserved register to the operand for its live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range fails.

6. (Previously presented) The method of claim 1, wherein attempting to allocate the scratch register to the operand for its live range comprises:

attempting to deallocate a second scratch register from a tentative live range of a second operand when the attempt to allocate the scratch register to the operand for its live range fails;

allocating the second scratch register to the operand for its live range when the second scratch register is successfully deallocated from the second operand for its tentative live range; and

attempting to allocate the preserved register to the operand for its live range when the attempt to deallocate the second scratch register from the second operand for its tentative live range fails.

7. (Previously presented) The method of claim 1, further comprising:

marking the live range of the operand as a tentative live range when current information about the live range is insufficient to determine whether or not the live range spans the function call;

pre-allocating both the scratch register and the preserved register to the operand for its tentative live range;

processing one or more additional instructions; and

waiting for additional information about the tentative live range.

8. (Previously presented) The method of claim 7, further comprising:

receiving the additional information about the tentative live range;

deallocating the preserved register from the operand for its tentative live range when the last use of the operand for its tentative live range has been scheduled; and

deallocating the scratch register from the operand for its tentative live range when the function call is scheduled before the last use of the operand for its tentative live range or when the scratch register is reallocated by any other process.

9. (Previously presented) The method of claim 7, further comprising:

attempting to allocate a currently unused preserved register in the list of previously used preserved registers to the operand for its tentative live range; and attempting to allocate a new unused preserved register to the operand for its tentative live range when the attempt to allocate the currently unused preserved register to the operand for its tentative live range fails.

10. (Previously presented) The method of claim 7, further comprising:

attempting to allocate the preserved register to the operand for its tentative live range when the scratch register is determined to be unavailable.

11. (Previously presented) The method of claim 1, further comprising:

inserting a register spill when the attempt to allocate the preserved register and the attempt to allocate the scratch register to the operand for its live range both fail; and

either re-attempting to allocate the preserved register and the scratch register to the operand for its live range, or selecting a second instruction to schedule.

12. (Previously presented) An article of manufacture comprising a machine readable medium having logic thereon that, when executed, the logic provides for processing a group of instructions, the logic further providing for:

selecting an instruction to schedule, the instruction associated with an operand, wherein the operand comprises a live range representing the operational duration of the operand relative to the group of instructions being processed;

attempting to allocate a preserved register to the operand for its live range when its live range spans a function call;

attempting to allocate a scratch register to the operand for its live range when its live range does not span the function call; and

scheduling the instruction when the allocation of the preserved register or the scratch register is successful.

13. (Previously presented) The article of manufacture of claim 12, the logic further providing for:

terminating the attempt to allocate the preserved register or the scratch register to the operand for its live range when the operand's use by the selected instruction falls at the end of the live range; and

marking the preserved register or the scratch register allocated to the operand as available for reallocation.

14. (Previously presented) The article of manufacture of claim 12, wherein the logic providing for attempting to allocate the preserved register comprises:

allocating the preserved register to the operand for its live range when the preserved register is in a list of previously used preserved registers marked as available for allocation.

15. (Previously presented) The article of manufacture of claim 14, the logic further providing for:

attempting to find a second preserved register in the list of previously used preserved registers and currently allocated to a tentative live range of a second operand; and

reallocating the second preserved register to the operand for its live range and reserving a different preserved register to the second operand for its tentative live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range is successful.

16. (Previously presented) The article of manufacture of claim 15, the logic further providing for:

adding a new preserved register to the list of previously used preserved registers; and

allocating the new preserved register to the operand for its live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range fails.

17. (Previously presented) The article of manufacture of claim 12, wherein the logic providing

for attempting to allocate the scratch register to the operand for its live range comprises:

attempting to deallocate a second scratch register from a tentative live range of a second operand when the attempt to allocate the scratch register to the operand for its live range fails;

allocating the second scratch register to the operand for its live range when the second scratch register is successfully deallocated from the second operand for its tentative live range; and

attempting to allocate the preserved register to the operand for its live range when the attempt to deallocate the second scratch register from the second operand for its tentative live range fails.

18. (Previously presented) The article of claim 12, the logic further providing for:

marking the live range of the operand as a tentative live range when current information about the live range is insufficient to determine whether or not the live range spans the function call;

pre-allocating both the scratch register and the preserved register to the operand for its tentative live range;

processing one or more additional instructions; and

waiting for additional information about the tentative live range.

19. (Previously presented) The article of manufacture of claim 18, the logic further providing for:

receiving the additional information about the tentative live range;

deallocating the preserved register from the operand for its tentative live range when the last use of the operand for its tentative live range has been scheduled; and

deallocating the scratch register from the operand for its tentative live range when the function call is scheduled before the last use of the operand for its tentative live range or when the scratch register is reallocated by any other process.

20. (Previously presented) The article of manufacture of claim 18, the logic further providing for:

attempting to allocate a currently unused preserved register in the list of previously used preserved registers to the operand for its tentative live range; and

attempting to allocate a new unused preserved register to the operand for its tentative live range when the attempt to allocate the currently unused preserved register to the operand for its tentative live range fails.

21. (Previously presented) The article of manufacture of claim 18, the logic further providing for:

attempting to allocate the preserved register to the operand for its tentative live range when the scratch register is determined to be unavailable.

22. (Previously presented) The article of manufacture of claim 12, the logic further providing for:

inserting a register spill when the attempt to allocate the preserved register and the attempt to allocate the scratch register to the operand for its live range both fail; and

either re-attempting to allocate the preserved register and the scratch register to the operand for its live range, or selecting a second instruction to schedule.

23. (Previously presented) An apparatus comprising:

an instruction scheduler to process a group of instructions, the instruction scheduler to further select an instruction to schedule, the instruction associated with an operand, the operand comprising a live range to represent the operational duration of the operand relative to the group of instructions to be processed; and

a register allocator to attempt to allocate a preserved register to the operand for its live range when its live range spans a function call and attempt to allocate a scratch register to the operand for its live range when its live range does not span the function call.

24. (Previously presented) The apparatus of claim 23, wherein the register allocator to further

terminate the attempt to allocate the preserved register or the scratch register to the operand for its live range when the operand's use by the selected instruction is to fall at the end of the live range, and

mark the preserved register or the scratch register to be allocated to the operand as available for reallocation.

25. (Previously presented) The apparatus of claim 23, wherein the register allocator to further

allocate the preserved register to the operand for its live range when the preserved register is in a list of previously used preserved registers to be marked as available for allocation.

26. (Previously presented) The apparatus of claim 25, wherein the register allocator to further

attempt to find a second preserved register in the list of previously used preserved registers to be allocated to a tentative live range of a second operand,

reallocate the second preserved register to the operand for its live range, and

reserve a different preserved register to the second operand for its tentative live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range is successful.

27. (Previously presented) The apparatus of claim 26, wherein the register allocator to further

add a new preserved register to the list of previously used preserved registers and

allocate the new preserved register to the operand for its live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range fails.

28. (Previously presented) The apparatus of claim 23, wherein the attempt to allocate the scratch register to the operand for its live range comprises the register allocator to further:

attempt to deallocate a second scratch register from a tentative live range of a second operand when the attempt to allocate the scratch register to the operand for its live range fails,

allocate the second scratch register to the operand for its live range when the second scratch register is successfully deallocated from the second operand for its tentative live range, and

attempt to allocate the preserved register to the operand for its live range when the attempt to deallocate the second scratch register from the second operand for its tentative live range fails.

29. (Previously presented) The apparatus of claim 23, wherein the register allocator to further

pre-allocate both the scratch register and the preserved register to the operand for its tentative live range,

process one or more additional instructions, and

wait for additional information about the tentative live range.

30. (Previously presented) The apparatus of claim 29, wherein the register allocator to further

receive the additional information about the tentative live range,

deallocate the preserved register from the operand for its tentative live range when the last use of the operand for its tentative live range has been scheduled, and

deallocate the scratch register from the operand for its tentative live range when the function call is scheduled before the last use of the operand for its tentative live range or when the scratch register is reallocated by any other process.

31. (Previously presented) The apparatus of claim 30, wherein the register allocator to further

attempt to allocate an unused preserved register in the list of previously used preserved registers to the operand for its tentative live range, and

attempt to allocate a new unused preserved register to the operand for its tentative live range when the attempt to allocate the unused preserved register to the operand for its tentative live range fails.

32. (Previously presented) The apparatus of claim 29, wherein the register allocator to further attempt to allocate the preserved register to the operand for its tentative live range when the scratch register is determined to be unavailable.

33. (Previously presented) The apparatus of claim 23, the register allocator to further

insert a register spill when the attempt to allocate the preserved register and the attempt to allocate the scratch register to the operand for its live range both fail, and

either re-attempt to allocate the preserved register and the scratch register to the operand for its live range, or select a second instruction to schedule.

34. (Previously presented) The apparatus of claim 23, wherein the register allocator further comprises:

a scratch register allocator to allocate scratch registers to operands for their live ranges when the live ranges do not span a function call;

a preserved register allocator to allocate preserved registers to operands for their live ranges when the live ranges span the function call; and

a tentative register allocator to allocate either the scratch register or the preserved register to their respective live ranges when it is unknown whether or not the live ranges span the function call.

35. (Currently amended) A system comprising a processor to execute a compiler integrated with the system and a computer-readable storage medium having a group of instructions stored thereon, ~~an integrated compiler to compile a group of instructions,~~ wherein the ~~integrated~~ compiler integrated with the system comprises:

an instruction scheduler to process the group of instructions, the instruction scheduler to further select an instruction to schedule, the instruction associated with an operand, the operand comprising a live range to represent the operational duration of the operand relative to the group of instructions to be processed; and

a register allocator to attempt to allocate a preserved register to the operand for its live range when its live range spans a function call and attempt to allocate a scratch register to the operand for its live range when its live range does not span the function call.

36. (Previously presented) The system of claim 35, wherein the register allocator to further

terminate the attempt to allocate the preserved register or the scratch register to the operand for its live range when the operand's use by the selected instruction is to fall at the end of the live range, and

mark the preserved register or the scratch register to be allocated to the operand as available for reallocation.

37. (Previously presented) The system of claim 35, wherein the register allocator to further

allocate the preserved register to the operand for its live range when the preserved register is in a list of previously used preserved registers to be marked as available for allocation.

38. (Previously presented) The system of claim 37, wherein the register allocator to further

attempt to find a second preserved register in the list of previously used preserved registers to be allocated to a tentative live range of a second operand,

reallocate the second preserved register to the operand for its live range, and

reserve a different preserved register to the second operand for its tentative live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range is successful.

39. (Previously presented) The system of claim 38, wherein the register allocator to further

add a new preserved register to the list of previously used preserved registers and

allocate the new preserved register to the operand for its live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range fails.

40. (Previously presented) The system of claim 35, wherein the attempt to allocate the scratch register to the operand for its live range comprises the register allocator to further:

attempt to deallocate a second scratch register from a tentative live range of a second operand when the attempt to allocate the scratch register to the operand for its live range fails,

allocate the second scratch register to the operand for its live range when the second scratch register is successfully deallocated from the second operand for its tentative live range, and

attempt to allocate the preserved register to the operand for its live range when the attempt to deallocate the second scratch register from the second operand for its tentative live range fails.

41. (Previously presented) The system of claim 35, wherein the register allocator to further

pre-allocate both the scratch register and the preserved register to the operand for its tentative live range,

process one or more additional instructions, and

wait for additional information about the tentative live range.

42. (Previously presented) The system of claim 41, wherein the register allocator to further

receive the additional information about the tentative live range,

deallocate the preserved register from the operand for its tentative live range when the last use of the operand for its tentative live range has been scheduled, and deallocate the scratch register from the operand for its tentative live range when the function call is scheduled before the last use of the operand for its tentative live range or when the scratch register is reallocated by any other process.

43. (Previously presented) The system of claim 42, wherein the register allocator to further

attempt to allocate an unused preserved register in the list of previously used preserved registers to the operand for its tentative live range, and

attempt to allocate a new unused preserved register to the operand for its tentative live range when the attempt to allocate the unused preserved register to the operand for its tentative live range fails.

44. (Previously presented) The system of claim 41, wherein the register allocator to further

attempt to allocate the preserved register to the operand for its tentative live range when the scratch register is determined to be unavailable.

45. (Previously presented) The system of claim 35, the register allocator to further

insert a register spill when the attempt to allocate the preserved register and the attempt to allocate the scratch register to the operand for its live range both fail, and

either re-attempt to allocate the preserved register and the scratch register to the operand for its live range, or select a second instruction to schedule.

46. (Previously presented) The system of claim 35, wherein the register allocator further comprises:

a scratch register allocator to allocate scratch registers to operands for their live ranges when the live ranges do not span a function call;

a preserved register allocator to allocate preserved registers to operands for their live ranges when the live ranges span the function call; and a tentative register allocator to allocate either the scratch register or the preserved register to their respective live ranges when it is unknown whether or not the live ranges span the function call.